LISTING OF CLAIMS

1. - 9. (canceled)

10.   (currently amended) A <u>computer-implemented</u> method for applying XML-compatible markup to unstructured textual documents, <u>the method</u> comprising:

defining an XML schema in accordance with which documents are to be marked up;

opening a ~~given~~ target document in a host Application Programming Interface (API) enabled ~~generic~~ wordprocessor application capable of storing XML-compatible non-native markup ~~in its~~ documents;

using <u>an</u> ~~the~~ API of the ~~host~~ <u>wordprocessor</u> application to parse ~~the document~~ content <u>included in the target document</u> and to perform element pattern matching ~~and~~ <u>to</u> ~~yielding~~ <u>inferred</u> XML structure in accordance with the ~~chosen~~ <u>defined</u> XML schema; and

storing the inferred <u>XML</u> structure within the <u>target</u> document as XML-compatible markup via the API of the <u>wordprocessor</u> ~~host~~ application.

11.   (currently amended) A method as recited in claim 10 wherein said using step comprises ~~a structure inference method for parsing a given document;~~ recognizing instances of designated baseline elements via pattern search and matching; and inferring and constructing higher-level element structure in ~~best possible~~ <u>substantial</u> conformance with the defined XML schema.

12.   (currently amended) A method as recited in claim 10 wherein ~~the~~ original visual formatting and textual content of the target document remain intact after ~~applying XML markup to it~~ <u>storing the inferred XML structure within the target document as XML-compatible markup</u>.

13.   (currently amended) A method as recited in claim 10 <u>further comprising limiting</u> ~~wherein the~~ XML structure inference and markup creation ~~are limited~~ to a select range or number of select ranges of the target document.

14.   (currently amended) A method as recited in claim 10 <u>further comprising creating</u> ~~wherein~~ a structure inference definition for the ~~chosen~~ <u>defined</u> XML schema ~~is created~~ using ~~by means of~~ <u>a</u> dedicated Graphical User Interface (GUI) integrated in ~~the~~ <u>a</u> GUI workspace of the ~~host~~ <u>wordprocessor</u> application.

15.   (currently amended) A method as recited in claim 10 further comprising presenting ~~the~~ <u>a</u> user with a GUI ~~means~~ to review probable trouble spots in the <u>target</u> document and to manually correct and complete the automatically generated XML<u>-compatible</u> markup, <u>the</u> probable trouble spots comprising

unmarked ranges, missing required elements from the <u>defined</u> XML schema, and inferred XML structure being invalid according to the <u>defined XML</u> schema.

16.   (canceled)

17.   (new) A method as recited in claim 10 wherein opening a target document in a host Application Programming Interface (API) enabled wordprocessor application capable of storing XML-compatible non-native markup in documents includes opening the target document in a host API enabled wordprocessor application that includes a plug-in capable of storing XML-compatible markup.

18.   (new) A method as recited in claim 10 further comprising:
        identifying a target document type from a set of textual documents with generally consistent inherent logical structure and formatting;
        creating a structure inference definition for the target document type comprising a multiplicity of definitions of baseline elements, the baseline elements being select leaf-level or near-leaf-level elements from the target document type and having a schema context; and
        defining recognition patterns for the baseline elements.

19.   (new) A method as recited in claim 18 further comprising invoking a computer-executable engine to apply the structure inference definition to one or more instances of the target document type to produce XML structure relating to the defined schema, the operation of said engine comprising: parsing the one or more instances of the target document type.

20.   (new) A method as recited in claim 19 further comprising defining patterns and structure inference and construction rules for one or more levels of nested elements in a designated baseline element, and configuring the computer-executable engine to use said patterns and rules to produce nested element structure within a text range and the schema context of a baseline element.

21.   (new) A method as recited in claim 19 further comprising:
        deriving a state machine having transition labels by recursive aggregation of schema element content models, starting from a designated root element and moving to the level of designated baseline elements;
        incorporating identities and specific instances of baseline elements in the transition labels of the state machine; and
        configuring the computer-executable engine to compile and use the state machine to consider a relatively small number of expected baseline elements at a given document position.

22.   (new) A method as recited in claim 10 wherein opening a target document in a host Application Programming Interface (API) enabled wordprocessor application capable of storing XML-compatible non-native markup in documents includes detecting the target document in a predefined incoming document folder or receiving the target document via the API from an external client component.

23.   (new) A method as recited in claim 22 wherein using an API of the wordprocessor application to parse content included in the target document and to perform element pattern matching to yield inferred XML structure in accordance with the defined XML schema includes using the API of the wordprocessor application to automatically parse the content included in the target document and to perform element pattern matching to yield inferred XML structure in accordance with the defined XML schema after detecting the target document in a predefined incoming document folder or after receiving the target document via the API from the external client computer.

24.   (new) A method as recited in claim 22 further comprising creating a structure inference definition for the target document comprising a multiplicity of definitions of baseline elements, the baseline elements being select leaf-level or near-leaf-level elements from the second target document and having a schema context and defining recognition patterns for the baseline elements.

25.   (new) A method as recited in claim 10 wherein opening a target document in a host Application Programming Interface (API) enabled wordprocessor application capable of storing XML-compatible non-native markup in its documents includes opening multiple target documents.

26.   (new) A method as recited in claim 25 wherein using an API of the wordprocessor application to parse content included in the target document and to perform element pattern matching to yield inferred XML structure in accordance with the defined XML schema includes using the API of the wordprocessor application to parse content included in the multiple target documents sequentially or in parallel in an unattended batch mode.

27.   (new) A method as recited in claim 25 further comprising creating a structure inference definition for the multiple target documents comprising a multiplicity of definitions of baseline elements, the baseline elements being select leaf-level or near-leaf-level elements from the multiple documents and having a schema context and defining recognition patterns for the baseline elements.

28.   (new) A method as recited in claim 18 wherein creating a structure inference definition for the target document type comprising a multiplicity of definitions of baseline elements, the baseline elements being select leaf-level or near-leaf-level elements from the target document type and having a schema

context includes identifying a baseline element by a schema path comprising a sequence of one or more XML element or element type steps, a first one of the one or more XML element or element type steps designating a global schema element or type and each subsequent step designating a child element or element group of its predecessor.

29.   (new) A method as recited in claim 18 further comprising defining the recognition patterns for the baseline elements to include: text patterns selected from the group of literals, wildcards, and regular expressions; formatting patterns selected from the group of font style, font name, font size, composite style name, paragraph indentation, and outline level; and logical compositions of atomic text and formatting patterns and pattern groups.

30.   (new) A method as recited in claim 18 further comprising defining the recognition patterns for the baseline elements to include:

an optional leading pattern, intended to match a document range immediately preceding a content range of the baseline element, allowing intervening whitespace;

an optional content pattern, intended to match the content range of the baseline element; and

an optional trailing pattern, intended to match a document range immediately following the content range for the baseline element, allowing intervening whitespace, an end document position of the trailing pattern element serving as a starting position for matching recognition patterns of following baseline elements.

31.   (new) A method as recited in claim 18 wherein the defining of recognition patterns for the baseline elements comprises assigning a priority value or pattern weight value which influences a selection of one baseline element when the recognition patterns for more than one baseline element yield competing/ambiguous matches at a particular document position.